PCI-SIG ENGINEERING CHANGE NOTICE

TITLE:	Async Hot-Plug Updates
DATE:	Nov 29, 2018
AFFECTED DOCUMENT:	PCIe 4.0 Base Specification
SPONSORS:	Joe Cowan; Hewlett Packard Enterprise Austin Bolen; Dell EMC Dave Harriman; Intel Corporation

Part I

1. Summary of the Functional Changes

This ECR updates several areas related to hot-plug functionality, mostly related to async hot-plug, which is now growing in importance due to its widespread use with NVMe SSDs. All new functionality is optional.

1. Key problems with the existing Hot-Plug Surprise (HPS) mechanism are fixed for OS use, primarily by enabling presence detect (PD) control mechanisms to be driven exclusively by out-of-band PD, and using existing DLL Link Active control mechanisms in lieu of in-band PD.

2. Downstream Port Containment (DPC) is documented as the preferred mechanism for supporting the async hot-plug model, and the existing HPS mechanism is deprecated. Both mechanisms are still fully supported, and are documented with Implementation Notes using a common async hot-plug reference model.

3. A mechanism is added where a slot can support both HPS and DPC for async hot-plug, with the default being HPS. An OS desiring to use DPC can negotiate with system firmware to disable HPS so DPC can be used instead.

4. A new System Firmware Intermediary (SFI) Capability structure is defined, which system firmware can use to:

    a. Fix some key HPS problems transparently for existing OSs.

    b. Fix problems with “rogue Configuration Reads” not being blocked by OSs during certain critical periods (e.g., immediately following reset release during a hot add), resulting in undefined HW behavior. These can be fixed transparently for existing OSs.

    c. Support “Firmware First” functionality for system firmware to be notified of an async hot add, and be allowed to configure the newly added device before notifying the OS of it being added.

5. A subclass field is added to ERR_COR Messages, enabling Downstream Ports that send ERR_COR Messages to signal system firmware (“SIG_SFW”) to distinguish them from ERR_COR Messages used for other purposes. This restores the ability for ERR_COR Messages to be used for higher frequency purposes like reporting individual correctable Link errors while still using them to signal system firmware for relatively rare but high overhead firmware first services.

2. Benefits as a Result of the Changes

Key async hot-plug problems with HPS can be fixed either by an updated OS, or transparently for existing OSs by system firmware.

DPC is better enabled for use with async hot-plug, offering more robust hot-plug operation, as well as concurrently enabling robust Containment Error Recovery (CER) if supported by suitable SW/FW in the host.

Rare but critical problems with “rogue Configuration Reads” can be fixed by system firmware transparently to unmodified OSs, or by an OS if system firmware doesn’t support it.

System firmware can be notified of a newly added device and be given the opportunity to configure it before notifying the OS of it being added.

3. Assessment of the Impact

New HW features by default are disabled and backwards compatible with existing SW/FW. All new HW features are optional.

4. Analysis of the Hardware Implications

All of the new HW features are in Downstream Ports. Endpoint devices are not affected.

5. Analysis of the Software Implications

New HW features by default are disabled and backwards compatible with existing SW/FW.

Some new features can be utilized entirely by system firmware, and be transparent to existing OSs.

Some new features are intended for updated OS use

6. Analysis of the C&I Test Implications

This ECN allocates previously reserved bits and requires the creation of new tests for the new register fields. New C&I tests would be required if it is desirable to extend C&I coverage to explicitly evaluate these new features.

Part II

Detailed Description of the change

Change Terms and Acronyms section as follows:

Terms and Acronyms

…

Containment Error Recovery (CER)	A general error containment and recovery approach supported by Downstream Port Containment (DPC), where with suitable software/firmware support, many uncorrectable errors can be handled without disrupting applications.
orderly removal	A hot-plug removal model where the OS is notified when a user/operator wishes to remove an adapter, and the OS has the opportunity to prepare for the event (e.g., quiescing adapter activity) before granting permission for removal.

Change Section 2.2.8.3 as follows:

2.2.8.3 Error Signaling Messages

Error Signaling Messages are used to signal errors that occur on specific transactions and errors that are not necessarily associated with a particular transaction. These Messages are initiated by the agent that detected the error.

…

The initiator of the Message is identified with the Requester ID of the Message header. The Root Complex translates these error Messages into platform level events. Refer to Section 6.2 for details on uses for these Messages.

- ERR_COR Messages have an ERR_COR Subclass (ECS) field in the Message header that enables different subclasses to be distinguished from each other. See Figure 2-f1. ERR_NONFATAL and ERR_FATAL Messages do not have the ECS field.

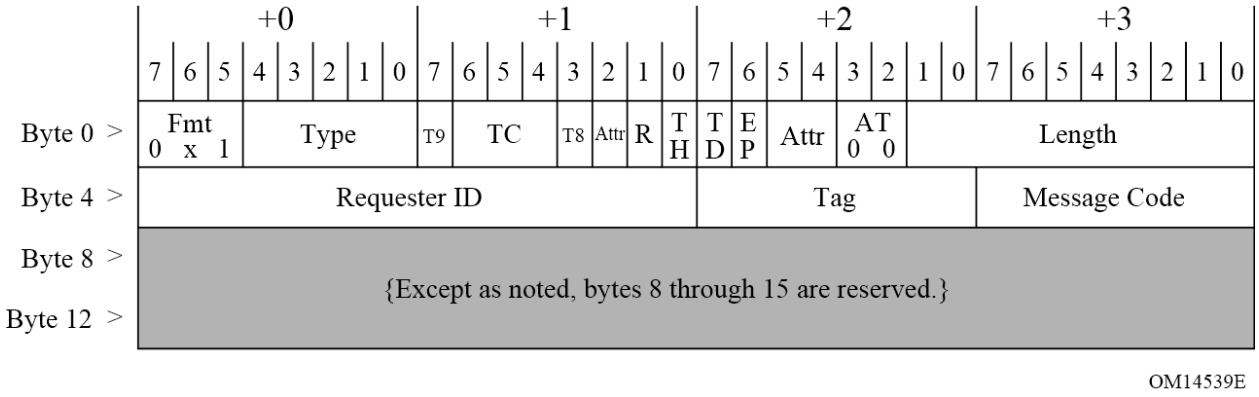

Figure 2-f1: ERR_COR Message

Note to Editor: Please create a new figure by leveraging the preceding existing figure to add a 2-bit ECS field in bits 7:6 of byte 8, and indicating the remaining bits as Reserved. Also indicate a Fmt field value of 001, a Type field value of 1000, a TC field value of 000, a TH bit indication of R, an Attr indication of R R, a Length field indication of Reserved, and a Message Code value of 0011 0000.

- ❑ The ERR_COR Subclass (ECS) field is encoded as shown in Table 2-t1, indicating the ERR_COR Message subclass.

Table 2-t1: ERR_COR Subclass (ECS) Field Encodings

ECS coding	Description
00	**ECS Legacy** – The value inherently used if a Requester does not support ECS capability. ECS-capable Requesters must not use this value. See Section 7.5.3.3.
01	**ECS SIG_SFW** – Must be used by an ECS-capable Requester when signaling a DPC or SFI event with an ERR_COR Message.
10	**ECS SIG_OS** – Must be used by an ECS-capable Requester when signaling an AER or RP PIO event with an ERR_COR Message.
11	**ECS Extended** – Intended for possible future use. Requesters must not use this value. Receivers must handle the signal internally the same as ECS SIG_OS.

Change Section 6.2.2.1 Correctable Errors as follows:

6.2.2.1 Correctable Errors

Correctable errors include those error conditions where hardware can recover without any loss of information. Hardware corrects these errors and software intervention is not required. For example, an LCRC error in a TLP that might be corrected by Data Link Level Retry is considered a correctable error. Measuring the frequency of Link-level correctable errors may be helpful for profiling the integrity of a Link.

Correctable errors also include transaction-level cases where one agent detects an error with a TLP, but another agent is responsible for taking any recovery action if needed, such as re-attempting the operation with a separate subsequent transaction. The detecting agent can be

configured to report the error as being correctable since the recovery agent may be able to correct it. If recovery action is indeed needed, the recovery agent must report the error as uncorrectable if the recovery agent decides not to attempt recovery.

The triggering of Downstream Port Containment (DPC) is not handled as an error, but it can be signaled as if it were a correctable error, since software that takes advantage of DPC can sometimes recover from the uncorrectable error that triggered DPC. See Section 6.2.10. An ERR_COR Message that's used for DPC signaling is intended to target system firmware, and may indicate so via the ERR_COR Subclass field.

Similarly, ERR_COR may be used by the System Firmware Intermediary (SFI) capability to signal system firmware, and must indicate so via the ERR_COR Subclass field. See Section 6.7.4.

Change Section 6.2.6 Error Message Controls as follows:

6.2.6 Error Message Controls

Error Messages have a complex set of associated control and status bits. Figure 6-3 provides a ~~conceptual~~ high-level summary in the form of a pseudo logic diagram for how error Messages are generated, logged, forwarded, and ultimately notified to the system. Not all control and ~~logged~~ status bits are shown. The logic gates shown in this diagram are intended for conveying general concepts, and not for direct implementation.

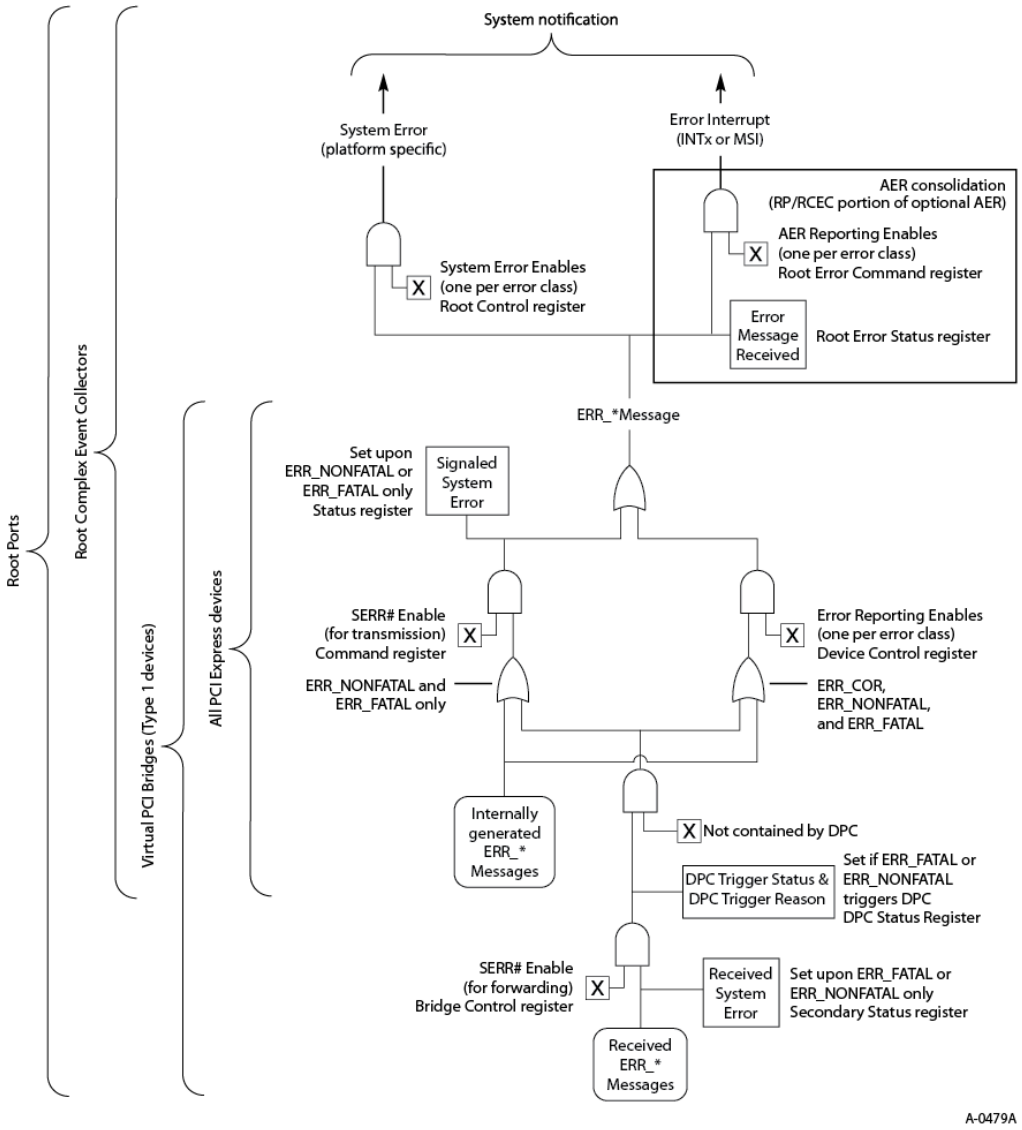

Figure 6-3: Pseudo Logic Diagram for Selected Error Message Control and Status Bits

Change Section 6.2.10 Downstream Port Containment (DPC) as follows:

6.2.10 Downstream Port Containment (DPC)

Downstream Port Containment (DPC) is an optional normative feature of a Downstream Port. DPC halts PCI Express traffic below a Downstream Port after an unmasked uncorrectable error is detected at or below the Port, avoiding the potential spread of any data corruption, and ~~permitting error recovery~~ supporting Containment Error Recovery (CER) if ~~supported~~ implemented by software. A Downstream Port indicates support for DPC by implementing a DPC Extended Capability structure, which contains all DPC control and status bits. See Section 7.9.15.

…

IMPLEMENTATION NOTE

Selecting Non-Posted Request Response During DPC

The DPC Completion Control bit determines how a Downstream Port responds to a Non-Posted Request (NPR) received during DPC. The selection needs to take into account how the rest of the platform handles ~~PCI Express uncorrectable error recovery~~ Containment Error Recovery (CER).

While specific ~~PCI Express uncorrectable error recovery mechanisms~~ CER policy details in a platform are outside the scope of this specification, here are some guidelines based on general considerations.

If the platform or drivers do not support ~~a PCI Express uncorrectable error recovery strategy~~ CER policies, ~~there's no envisioned benefit to enabling DPC, and thus no need to select the NPR response~~ it's recommended to select UR Completions, which is the standard behavior when a device is not present.

If the ~~PCI Express uncorrectable error recovery~~ CER strategy relies on software detecting containment by looking for all 1's returned by PIO reads, then a UR Completion may be the more appropriate selection, assuming the RP synthesizes an all 1's return value for PIO reads that return UR Completions. The all 1's synthesis would need to occur for PIO reads that target Configuration Space, Memory Space, and perhaps I/O Space.

If the ~~PCI Express uncorrectable error recovery~~ CER strategy utilizes a mechanism that handles UR and CA Completions differently for PIO reads, then a CA Completion might be the more appropriate selection. CA Completions coming back from a ~~PCI Express~~ PCIe device normally indicate a device programming model violation, which may need to trigger ~~Root~~ Port containment and error recovery.

…

6.2.10.2 DPC ERR_COR Signaling

A DPC-capable Downstream Port must support ERR_COR signaling, independent of whether it supports Advanced Error Reporting (AER) or not. DPC ERR_COR signaling is enabled by the DPC ERR_COR Enable bit in the DPC Control register. DPC triggering is indicated by the DPC Trigger Status bit in the DPC Status register. DPC ERR_COR signaling is managed independently of DPC interrupts, and it is permitted to use both mechanisms concurrently.

If the DPC ERR_COR Enable bit is Set, and the Correctable Error Reporting Enable bit in the Device Control register or the DPC SIG_SFW Enable bit in the DPC Control register is Set, the Port must send an ERR_COR Message each time the DPC Trigger Status bit transitions from Clear to Set. DPC ERR_COR signaling must not Set the Correctable Error Detected bit in the Device Status register, since this event is not handled as an error. If the Downstream Port supports ERR_COR Subclass capability, this DPC ERR_COR signaling event must set the DPC SIG_SFW Status bit in the DPC Status register and also set the ERR_COR Subclass field in the ERR_COR Message to indicate ECS SIG_SFW.

…

IMPLEMENTATION NOTE

Use of DPC ERR_COR Signaling

It is recommended that operating systems use DPC interrupts for signaling when DPC has been triggered. While DPC ERR_COR signaling indicates the same event, DPC ERR_COR signaling is primarily intended for use by ~~platform~~ system firmware, when it needs to be notified in order to do its own logging of the event or provide ~~"~~firmware first~~"~~ services.

…

6.2.10.4 Software Triggering of DPC

…

IMPLEMENTATION NOTE

Avoid Disable Link and Hot-Plug Surprise Use With DPC

It is recommended that software not Set the Link Disable bit in the Link Control register while DPC is enabled but not triggered. Setting the Link Disable bit will cause the Link to be directed to DL_Down, invoking some semantics similar to those in DPC, but lacking others. ~~The~~ If DPC is enabled, the subsequent arrival of any Posted Requests will likely trigger DPC ~~soon~~, anyway. If DPC is enabled, the recommended method for software to disable the Link is to write a 1b to the optional DPC Software Trigger bit in the DPC Control register. If the DPC Software Trigger bit is not implemented, software should disable DPC and use Disable Link instead. If the operating system is performing this action, but DPC is owned by system firmware, the operating system should coordinate disabling DPC with system firmware.

~~Similarly, it is recommended that a Port that supports DPC not Set the Hot-Plug Surprise bit in the Slot Capabilities register.~~ DPC is not recommended for use concurrently with the Hot-Plug Surprise mechanism, indicated by the Hot-Plug Surprise bit in the Slot Capabilities register being Set. Having this bit Set blocks the reporting of Surprise Down errors, preventing DPC from being triggered by this important error, greatly reducing the benefit of DPC. See Section 6.7.4.4 for guidance on slots supporting both mechanisms.

6.2.10.5 DL_Active ERR_COR Signaling

Support for this feature is indicated by the DL_Active ERR_COR Signaling Supported bit in the DPC Capability register. The feature is enabled by the DL_ACTIVE ERR_COR Enable bit in the DPC Control register. The DL_ACTIVE state is indicated by the Data Link Layer Link Active bit in the Link Status register. DL_ACTIVE ERR_COR signaling is managed independently of Data Link Layer State Changed interrupts, and it is permitted to use both mechanisms concurrently.

If the DL_ACTIVE ERR_COR Enable bit is Set, and the Correctable Error Reporting Enable bit in the Device Control register or the DPC SIG_SFW Enable bit in the DPC Control register is Set, the Port must send an ERR_COR Message each time the Link transitions into the

DL_ACTIVE state. DL_ACTIVE ERR_COR signaling must not Set the Correctable Error Detected bit in the Device Status register, since this event is not handled as an error. If the Downstream Port supports ERR_COR Subclass capability, this DPC ERR_COR signaling event must set the DPC SIG_SFW Status bit in the DPC Status register and also set the ERR_COR Subclass field in the ERR_COR Message to indicate ECS SIG_SFW. In contrast to Data Link Layer State Changed interrupts, DL_Active ERR_COR signaling only indicates the Link enters the DL_Active state, not when the Link exits the DL_Active state.

…

IMPLEMENTATION NOTE

Use of DL_ACTIVE ERR_COR Signaling

It is recommended that operating systems use Data Link Layer State Changed interrupts for signaling when DL_ACTIVE changes state. While DL_ACTIVE ERR_COR signaling indicates a subset of the same events, DL_ACTIVE ERR_COR signaling is primarily intended for use by ~~platform~~ system firmware, when it needs to be notified in order to do Downstream Port configuration or provide ~~"~~firmware first~~"~~ services.

Change Section 6.7 PCI Express Hot-Plug Support as follows:

6.7 PCI Express Native Hot-Plug ~~Support~~

The PCI Express architecture is designed to natively support both hot-add and hot-removal ("hot-plug") of cables, add-in cards, and modules. PCI Express native hot-plug ~~support~~ provides a "toolbox" of mechanisms that allow different user/operator models to be supported using a self-consistent infrastructure. These mechanisms may be used to implement orderly addition/removal that relies on coordination with the operating system (e.g., traditional PCI hot-plug), as well as async removal, which proceeds without lock-step synchronization with the operating system. This section defines the set of hot-plug mechanisms and specifies how the elements of hot-plug, such as indicators and push buttons, must behave if implemented in a system.

…

6.7.2 Registers Grouped by Hot-Plug Element Association

The registers described in this section are grouped by hot-plug element to convey all registers associated with implementing each element. Registers associated with each Downstream Port implementing a hot-plug capable slot are located in the Device Capabilities, Slot Capabilities, Slot Control, ~~and~~ Slot Status, and Slot Capabilities 2 registers~~, and Slot Capabilities 2~~ in the PCI Express Capability structure (see Section 7.6.3). Registers reporting the presence of hot-plug elements associated with the device Function on an adapter are located in the Device Capabilities register (also in the PCI Express Capability structure).

…

6.7.2.5 Presence Detect Registers

In-Band PD Disable Supported (Slot Capabilities 2) – This bit indicates if the slot supports the disabling of in-band presence detect, which allows the out-of-band presence detect state to be reported independently of the in-band presence detect state.

In-Band PD Disable (Slot Control) – When Set, this bit disables the in-band presence detect mechanism from affecting the Presence Detect State bit, allowing that bit to be dedicated to reporting out-of-band presence detect.

Presence Detect State (Slot Status) – This bit indicates the presence of an adapter in the slot.

…

6.7.2.9 Port Capabilities and Slot Information Registers

…

Hot-Plug Surprise (Slot Capabilities) – When Set, this bit indicates ~~that adapter removal from the system without any prior notification is permitted for the associated form factor~~ that the Hot-Plug Surprise mechanism for handling async removal is enabled for this slot. See Section 6.7.6.

…

6.7.4 System Firmware Intermediary (SFI) Support

The System Firmware Intermediary (SFI) Capability is an optional normative feature of a Downstream Port. Some SFI functionality is focused on hot-pluggable slots, as indicated by the Hot-Plug Capable bit in the Slot Capabilities register being Set, while some SFI functionality is useful outside that context. If a Downstream Port supports an SFI Capability structure, the following bits must be Set:

- ❑ Data Link Layer Link Active Reporting Capable bit in the Link Capabilities register
- ❑ DRS Supported bit in the Link Capabilities 2 register
- ❑ ERR_COR Subclass Capable bit in the Device Capabilities register

6.7.4.1 SFI ERR_COR Event Signaling

The SFI Capability has no support for generating INTx or MSI/MSI-X interrupts, since the capability is intended for use by system firmware.

A Downstream Port with SFI must support ERR_COR signaling, regardless of whether it supports Advanced Error Reporting (AER) or not. SFI ERR_COR event signaling is enabled independently by the SFI OOB PD Changed Enable, SFI DLL State Changed Enable, and SFI DRS Signaling Enable bits in the SFI Control register. These events are indicated by the SFI OOB PD Changed, SFI DLL State Changed, and SFI DRS Received bits in the SFI Status register.

If the Correctable Error Reporting Enable bit in the Device Control register is Set, the Port must send an ERR_COR Message each time one of the enabled conditions becomes satisfied. SFI

ERR_COR event signaling must not Set the Correctable Error Detected bit in the Device Status register, since this event is not handled as an error.

IMPLEMENTATION NOTE

ERR_COR Signaling for DPC DL_Active vs. SFI DLL State Changed

DPC implements ERR_COR signaling for DL_Active, whereas SFI implements ERR_COR signaling for SFI DLL State Changed, which are related but non-identical conditions. The DL_Active condition occurs when the Data Link Layer Link Active bit in the Link Status register changes from 0b to 1b, and this bit can be masked by the SFI DLL State Mask bit in the SFI Control register. The SFI DLL State Changed condition occurs when the SFI DLL State bit in the SFI Status register changes its value either by becoming Set or becoming Clear, and this condition is always based on the actual Data Link Layer state.

6.7.4.2 SFI Downstream Port Filtering (DPF)

Downstream Port Filtering (DPF) is a mechanism where a Downstream Port can handle specified Request TLPs that target Components below it as if the Link is in DL_Down. See Section 2.9.1.

DPF has two modes of filtering Request TLPs that target Components below the Downstream Port. The first mode filters all such Request TLPs; the second mode filters only Configuration Request TLPs. Other TLPs must not be filtered or blocked by DPF.

One key use case for DPF is guaranteeing that asynchronous system software activities like bus scans do not unintentionally send Configuration Requests to devices that are not yet ready following a Conventional Reset, since such accesses result in undefined hardware behavior. See Section 6.6.1.

Another key use case for DPF is supporting "FW First" functionality, enabling system firmware, when notified of an async hot add, to configure the newly added device before making the device visible to the operating system. For this use case, the SFI CAM mechanism enables the Downstream Port itself to generate Configuration Request TLPs targeting Downstream Components, and those TLPs are not filtered or blocked by the DPF mechanism. See Sections 6.7.4.3, 7.9.21.5, and 7.9.21.6.

6.7.4.3 SFI CAM

The SFI Configuration Access Method (CAM) provides a means for SFI-aware system firmware to have the Downstream Port proxy (pass through) Configuration Requests targeting Components below the Downstream Port when DPF is enabled. The SFI CAM is always enabled.

To use the SFI CAM, software first writes to the SFI CAM Address register, specifying the target Configuration address. Software then reads or writes the SFI CAM Data register to cause a proxied Configuration Request to be generated and transmitted to the Downstream Component.

The following rules apply:

- All TLP fields used for the proxied Configuration Request are identical to those in the Configuration Request that targeted the SFI CAM Data register, with the following exceptions:
  - The target Bus Number, Device Number, and Function Number come from the SFI CAM Address register.
  - The Extended Register Number and Register Number come from the SFI CAM Address register.
  - The LCRC is regenerated.
  - If present, the ECRC is regenerated.
- The SFI CAM must not apply the Completion Timeout mechanism to the Request.
- System firmware must ensure that between the time it writes to the SFI CAM Address register and its subsequent read or write of the SFI CAM Data register completes, no other threads modify the SFI CAM Address register; otherwise, the result is undefined.
- If there is a detected error associated with the proxied Configuration Request, this is a reported error associated with the Downstream Port implementing the SFI CAM (see Section 6.2).
- Completions flowing Upstream must be passed through the Downstream Port unmodified.

IMPLEMENTATION NOTE

Serialized Use of the SFI CAM Address and Data Registers

As described above, system firmware must ensure that between the time it writes to the SFI CAM Address register and its subsequent read or write of the SFI CAM Data register completes, no other threads modify the SFI CAM Address register. For example, a semaphore or other synchronization mechanism can be used to ensure this serialization.

For platforms where a processor store instruction to Configuration Space is effectively posted, software must still ensure that the resulting Configuration Write completes before another software thread modifies the SFI CAM Data register. On such platforms, the mechanism for determining when a Configuration Write completes is platform specific.

Given appropriate serialization, the SFI CAM works correctly with Configuration Requests that result in CRS Completions, even when the Root Complex automatically re-issues the Configuration Request as a new Request. The re-issued Configuration Request will again be sent to the SFI CAM Data register, and the associated Downstream Port will again generate a Configuration Request targeting the Downstream Component. As long as the SFI CAM Address register isn't modified by other software until the Configuration Request completes, the sequence can repeat indefinitely until a non-CRS Completion is returned or a Completion Timeout occurs.

When CRS Software Visibility is enabled, the SFI CAM still works correctly with Configuration Requests that result in CRS Completions. Any Completions with a CRS Completion Status flow back to the original Requester, which handles them as required by CRS Software Visibility semantics. See Section 2.3.2.

IMPLEMENTATION NOTE

Use of Assigned Bus Numbers with the SFI CAM

When a Downstream Port has DPF enabled, the SFI CAM can be used by SFI-aware system firmware to configure and access the sub-hierarchy below the Port without other software being able to do so. While the Bus Number configuration below the Port is generally not visible to other software, Bus Numbers configured for use below the Port should be limited to those already assigned to the Port since TLPs coming Upstream through the Port may contain IDs with the configured Bus Numbers. If any errors are detected and logged with those TLPs, the Bus Numbers can become visible to other software, creating confusion if they overlap with Bus Numbers used elsewhere in the system.

6.7.4.4 SFI Interactions with Readiness Notifications

The SFI Capability is able to mask the reporting of received Device Readiness Status (DRS) Messages as well as emulate them being received. This functionality is useful when SFI's Downstream Port Filtering (DPF) mechanism is being used to block operating system visibility of a device or sub-hierarchy below the Downstream Port.

Rules:

- When the SFI DRS Mask bit is Set, the DRS Message Received bit in the Link Status 2 register value must be 0b.
- The SFI DRS Received bit must always indicate the actual state of the DRS Message Received condition.
- When the SFI DRS Mask bit is Clear and a 1b is written to the SFI DRS Trigger bit, the Downstream Port must behave as if a DRS Message was received.

IMPLEMENTATION NOTE

SFI Transparent Optimizations for Device Readiness

Certain devices may need more time to become Configuration-Ready following a hot-add operation than permitted. See Section 6.6.1.

If system firmware is aware of such devices, it can use the SFI DPF mechanism to block operating system visibility of a newly added device, wait the necessary amount of time for the device to become Configuration-Ready, and then expose the device to the operating system.

To avoid the operating system from unnecessarily waiting additional time for the newly exposed device to become Configuration-Ready, system firmware can use the SFI DRS Trigger bit to have the Downstream Port emulate the reception of a DRS Message. An operating system that supports DRS can then immediately discover and configure the newly exposed device.

The newly exposed device doesn't necessarily need to be DRS capable itself. Since an Upstream Port is expressly permitted to send DRS Messages even when its DRS Supported bit is Clear, the Downstream Port above it can legitimately emulate receiving a DRS Message from it even if it is incapable of sending DRS Messages.

It should also be noted that in cases where system firmware is aware of a device becoming Configuration-Ready early, system firmware can expose this to the operating system using the SFI DRS Trigger mechanism.

Although SFI is not intended to be used by operating system software, it is recommended that operating systems used in platforms supporting SFI implement support for DRS, so that the system as a whole can have the benefits of this optimized Device Readiness timing.

IMPLEMENTATION NOTE

SFI DPF and Function Readiness Status (FRS) Messages

Downstream Port Filtering (DPF) does not affect the generation or propagation of FRS Messages. No FRS Messages are generated by a device when it becomes ready as part of an async hot-add operation. However, if system firmware performs operations on a device that result in FRS events, the resulting FRS Messages may be visible to the operating system. See Sections 2.2.8.6.4 and 6.23.2.

6.7.4.5 SFI Suppression of Hot-Plug Surprise Functionality

If a slot supports Hot-Plug Surprise (HPS) functionality as indicated by the Hot-Plug Surprise bit in the Slot Capability register being Set, the SFI HPS Suppress bit in the SFI Control register can be used to force the Hot-Plug Surprise bit to be Clear, and disable the associated Hot-Plug Surprise functionality.

HPS suppression is useful when a Downstream Port / slot combination supports both HPS and Downstream Port Containment (DPC). DPC is not recommended for concurrent use with HPS, so if a slot has HPS capability enabled, DPC should not be enabled. If software wishes to use DPC, software should first Set the SFI HPS Suppress bit in order to disable HPS functionality, allowing DPC to function properly.

IMPLEMENTATION NOTE

Software Negotiation of Hot-Plug Surprise Functionality

Assuming that system firmware owns the SFI Capability structure, it is recommended that for backward compatibility with older operating systems, Hot-Plug Surprise functionality be enabled by default on slots supporting async removal. Then, if the slot also supports DPC and the operating system wishes to use it instead, the operating system will request that HPS be suppressed by system firmware, and system firmware will determine whether to Set or Clear the SFI HPS Suppress bit.

~~6.7.4~~6.7.5 Firmware Support for Hot-Plug

Some systems that include hot-plug capable Root Ports and Switches that are released before ACPI-compliant operating systems with native hot-plug support are available, can use ACPI firmware for propagating hot-plug events. Firmware control of the hot-plug registers must be

disabled if an operating system with native support is used. Platforms that provide ACPI firmware to propagate hot-plug events must also provide a mechanism to transfer control to the operating system. The details of this method are described in the *PCI Firmware Specification*.

~~6.7.5~~6.7.6 Async Removal

Async removal refers to the removal of an adapter or disabling of a Downstream Port Link due to error containment without prior warning to the operating system. This is in contrast to ~~standard~~ orderly removal~~PCI hot-plug,~~ where removal operations are performed in a lock-step manner with the OS through a well-defined sequence of user actions and system management facilities. For example, the user presses the Attention Button to request permission from the OS to remove the adapter, but the user doesn't actually remove the adapter from the slot until the OS has quiesced activity to the adapter and granted permission for removal.

Since async removal proceeds before the rest of the PCI Express hierarchy or OS necessarily becomes aware of the event, special consideration is required beyond that needed for standard PCI hot-plug. This section outlines PCI Express events that may occur as a side effect of async removal and mechanisms for handling async removal.

Since async removal may be unexpected to both the Physical and Data Link Layers of the Downstream Port associated with the slot, Correctable Errors may be reported as a side effect of the event (i.e. Receiver Error, Bad TLP, and Bad DLLP). If these errors are reported, software should handle them as an expected part of this event.

Requesters may experience Completion Timeouts associated with Requests that were accepted, but will never be completed by removed Completers. Any resulting Completion Timeout errors in this context should be handled as an expected part of this event.

Async removal may result in a transition from DL_Active to DL_Down in the Downstream port. This transition may result in a Surprise Down error. In addition, Requesters in the PCI Express hierarchy domain may not become immediately aware of this transition and continue to issue Requests to removed Completers that must be handled by the Downstream Port associated with the slot.

Either Downstream Port Containment (DPC) or the Hot-Plug Surprise (HPS) mechanism may be used to support async removal as part of an overall async hot-plug architecture. See Appendix x for the associated reference model.

~~The Surprise Down error resulting from async removal may trigger Downstream Port Containment (See Section 6.2.10). Downstream Port Containment following an async removal may be utilized to hold the Link of a Downstream Port in the Disabled LTSSM state while host software recovers from the side effects of an async removal.~~

IMPLEMENTATION NOTE

Hot-Plug Surprise Mechanism Deprecated for Async Hot-Plug

The Hot-Plug Surprise (HPS) mechanism, as indicated by the Hot-Plug Surprise bit in the Slot Capabilities register being Set, is deprecated for use with async hot-plug. DPC is the recommended mechanism for supporting async hot-plug. See Section 6.7.4.4 for guidance on slots supporting both mechanisms.

With async removal, using HPS has serious downsides. Uncorrectable errors other than those that inherently bring down the Link need to be configured either to crash the system, be handled asynchronously by software, or be ignored. These include uncorrectable errors associated with Posted Memory Writes, TLPs with poisoned data, and Completion Timeouts. Uncorrectable errors ignored or handled asynchronously by software may make it impossible for the driver to determine which high-level operations complete successfully versus those that do not.

DPC provides a robust mechanism for supporting async removal. The TLP stream cleanly stops upon an uncorrectable error that triggers DPC. Operating System / driver stacks that support Containment Error Recovery (CER) can fully and transparently recover from many transient PCIe uncorrectable errors. DPC can support async removal and CER concurrently.

Change Section 7.3.3 Configuration Request Routing Rules as follows:

7.3.3 Configuration Request Routing Rules

…

For Root Ports, Switches, and PCI Express-PCI Bridges, the following rules apply:

- Propagation of Configuration Requests from Downstream to Upstream as well as peer-to-peer are not supported
  - Configuration Requests are initiated only by the Host Bridge, including those passed through the SFI CAM mechanism

Change Section 7.5.3 PCI Express Capability Structure as follows:

7.5.3 PCI Express Capability Structure

…

7.5.3.3 Device Capabilities Register (Offset 04h)

The Device Capabilities register identifies PCI Express device Function specific capabilities. Figure 7-25 details allocation of register fields in the Device Capabilities register; Table 7-19 provides the respective bit definitions.

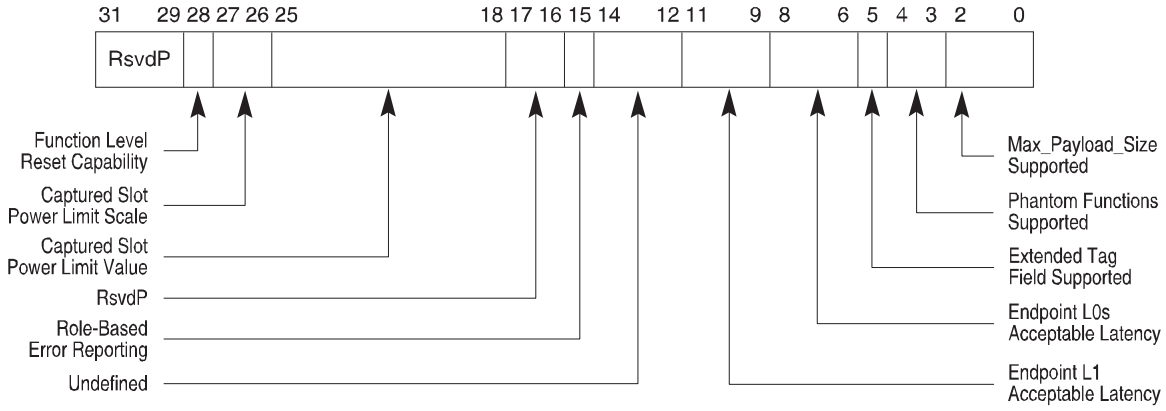

Figure 7-25: Device Capabilities Register

Note to Editor: Please update the preceding figure with the bit defined in the table below.

Table 7-19: Device Capabilities Register

Bit Location	Register Description	Attributes
…	…	…
16	**ERR_COR Subclass Capable** – When Set, this bit indicates that the Function supports the ERR_COR Subclass field in ERR_COR Messages, allowing different subclasses to be distinguished. See Section 2.2.8.3. Downstream Ports that implement the System Firmware Intermediary (SFI) capability must Set this bit. Downstream Ports that implement Downstream Port Containment (DPC) are strongly encouraged to Set this bit.	RO

7.5.3.10 Slot Control Register (Offset 18h)

…

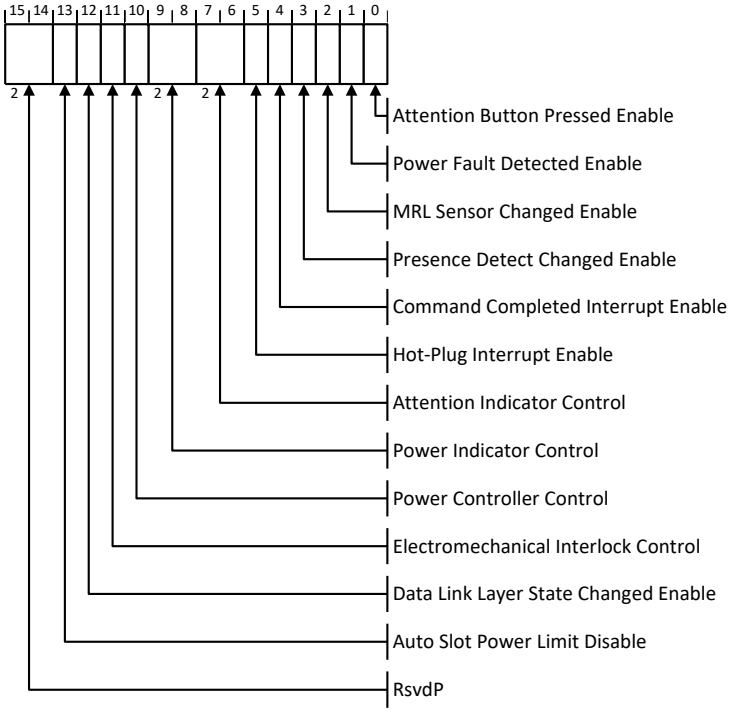

Figure 7-30: Slot Control Register

Note to Editor: Please update the preceding figure with the bit defined in the table below.

Table 7-25: Slot Control Register

Bit Location	Register Description	Attributes
…	…	…
14	**In-Band PD Disable –** When Set, this bit disables the in-band presence detect mechanism from affecting the Presence Detect State bit, allowing that bit to report out-of-band presence detect exclusively. Otherwise, the Presence Detect State bit reflects the logical OR of the in-band and out-of-band presence detect mechanisms. In addition, the In-Band PD Disable bit governs the Component Presence state for the Downstream Component Presence field in the Link Status 2 register. See Section 7.5.3.20. This bit must be implemented if the In-Band PD Disable Supported bit is 1b. Otherwise, this bit must be hardwired to 0b. Default value of this bit is 0b.	RW

…

7.5.3.11 Slot Status Register (Offset 1Ah)

…

Table 7-26: Slot Status Register

Bit Location	Register Description	Attributes
…	…	…
6	**Presence Detect State** – This bit indicates the presence of an adapter in the slot. When the In-Band PD Disable bit is Clear, this is reflected by the logical "OR" of the Physical Layer in-band presence detect mechanism and, if present, any out-of-band presence detect mechanism defined for the slot's corresponding form factor. Note that the in-band presence detect mechanism requires that power be applied to an adapter for its presence to be detected. Consequently, form factors that require a power controller for hot-plug must implement ~~a physical pin~~ an out-of-band presence detect mechanism. When the In-Band PD Disable bit is Set, the in-band presence detect mechanism has no effect on this bit. Defined encodings are: 0b ~~Slot Empty~~Adapter not Present 1b Adapter Present ~~in slot~~ This bit must be implemented on all Downstream Ports that implement slots. For Downstream Ports not connected to slots (where the Slot Implemented bit of the PCI Express Capabilities register is 0b), this bit must be hardwired to 1b.	RO

…

7.5.3.20 Link Status 2 Register (Offset 32h)

…

Table 7-35: Link Status 2 Register

Bit Location	Register Description	Attributes
…	…	…
14:12	**Downstream Component Presence** – This field indicates the presence and DRS status for the Downstream Component, if any, connected to the Link; defined values are: 000b Link Down – Presence Not Determined	RO/RsvdZ

Bit Location	Register Description	Attributes
	001b Link Down – Component Not Present indicates the Downstream Port (DP) has determined that a Downstream Component is not present 010b Link Down – Component Present indicates the DP has determined that a Downstream Component is present, but the Data Link Layer is not active 011b Reserved 100b Link Up – Component Present indicates the DP has determined that a Downstream Component is present, but no DRS Message has been received since the Data Link Layer became active 101b Link Up – Component Present and DRS Received indicates the DP has received a DRS Message since the Data Link Layer became active 110b Reserved 111b Reserved Component Presence state must be determined by the logical "OR" of the Physical Layer in-band presence detect mechanism and, if present, any out-of-band presence detect mechanism implemented for the Link. If no out-of-band presence detect mechanism is implemented, then Component Presence state must be determined solely by the Physical Layer in-band presence detect mechanism. If the In-Band PD Disable bit in the Slot Control register is Set, the Physical Layer in-band presence detect mechanism must always indicate that no component is present. Component Presence, Link Up, and DRS Received states indicated by this field must reflect their maskable states, which are controlled by the SFI PD State Mask, SFI DLL State Mask, or SFI DRS Mask bits in the SFI Control register. See Section 7.9.21.3. This field must be implemented in any Downstream Port where the DRS Supported bit is Set in the Link Capabilities 2 register. This field is RsvdZ for all other Functions. Default value of this field is 000b.	

…

7.5.3.21 Slot Capabilities 2 Register (Offset 34h)

~~This section is a placeholder. There are no capabilities that require this register.~~

~~This register must be treated by software as RsvdP.~~

Note to Editor: Please create a figure here using standard PCIe conventions for a 32-bit RsvdP register with fields and/or bits as defined in the table below.

Figure 7-f1: Slot Capabilities 2 Register

Table 7-t1: Slot Capabilities 2 Register

Bit Location	Register Description	Attributes
0	**In-Band PD Disable Supported** – When Set, this bit indicates that this slot supports disabling the reporting of the in-band presence detect state, as controlled by the In-Band PD Disable bit in the Slot Control register. If the slot does not support an out-of-band presence detect mechanism, this bit must be Clear.	HwInit

Change Section 7.8.4 Advanced Error Reporting Capability as follows:

7.8.4 Advanced Error Reporting Capability

…

7.8.4.10 Root Error Status Register (Offset 30h)

The Root Error Status register reports status of error Messages (ERR_COR, ERR_NONFATAL, and ERR_FATAL) received by the Root Port, and of errors detected by the Root Port itself…

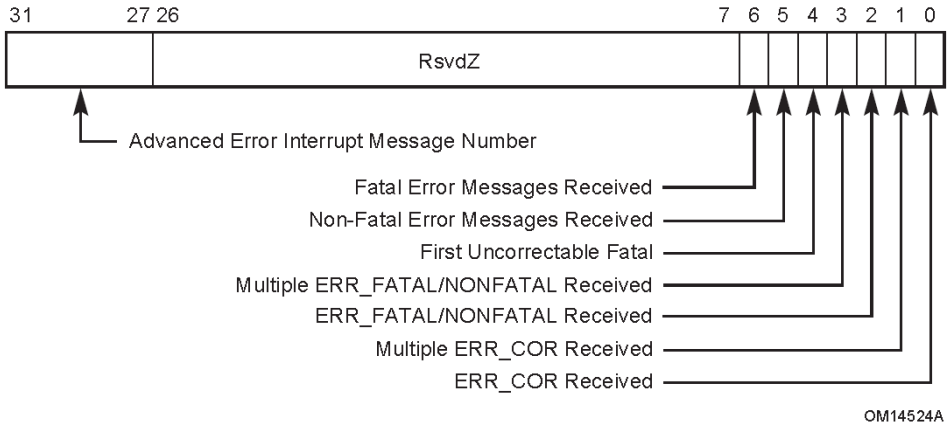

Figure 7-117: Root Error Status Register

Note to Editor: Please update the preceding figure with the field defined in the table below.

Table 7-95: Root Error Status Register

Bit Location	Register Description	Attributes
…	…	…
8:7	**ERR_COR Subclass** – If the Function is ERR_COR Subclass capable and the ERR_COR Received bit is not already Set, this field is loaded with the value of the ERR_COR Subclass field in the received ERR_COR Message. See Section 2.2.8.3. The value in this field is only valid when the ERR_COR Received bit is Set. If the Function is not ERR_COR Subclass capable, this field is Reserved. If the Function is ERR_COR Subclass capable and a SIG_SFW ERR_COR Message is received, system firmware should be signaled using a system-specific mechanism. Default value of this field is 00b.	ROS / RsvdZ

Change Section 7.9.15 DPC Extended Capability as follows:

7.9.15 DPC Extended Capability

…

If a Downstream Port implements the DPC Extended Capability, that Port must also be capable of reporting the DL_Active state, and indicate so by Setting the Data Link Layer Link Active Reporting Capable bit in the Link Capabilities register. See Section 7.6.3.6.

If a Downstream Port implements the DPC Extended Capability, it is strongly recommended for that Port to support ERR_COR Subclass capability, and indicate so by Setting the ERR_COR Subclass Capable bit in the Device Capabilities register. See Section 7.5.3.3.

7.9.15.3 DPC Control Register (Offset 06h)

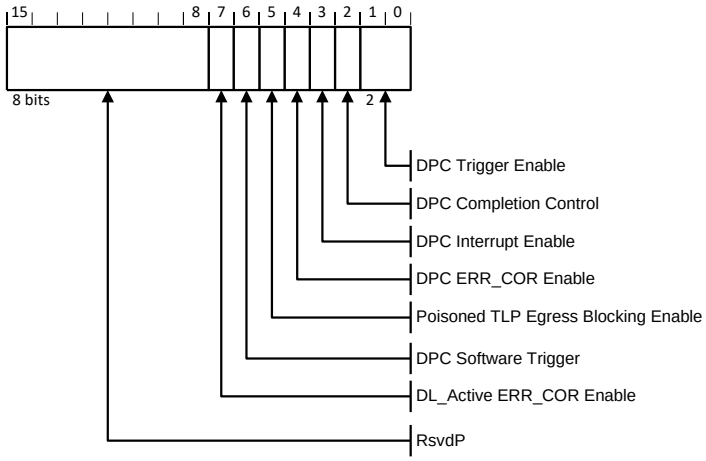

Figure 7-233: DPC Control Register

Note to Editor: Please update the preceding figure with the bit defined in the table below.

Table 7-188: DPC Control Register

Bit Location	Register Description	Attributes
…	…	…
8	**DPC SIG_SFW Enable –** This bit must be implemented if the ERR_COR Subclass Capable bit in the Device Capabilities register is Set; otherwise, it is permitted to be hardwired to 0b. If the ERR_COR Subclass Capable bit is Clear and software Sets this bit, the behavior is undefined. When Set, this bit enables sending an ERR_COR Message to indicate a DPC event that's been enabled for ERR_COR signaling. See Sections 6.2.10.2 and 6.2.10.5. This is an additional and alternative way to enable overall DPC ERR_COR signaling beyond the Correctable Error Reporting Enable bit in the Device Control register. This bit does not affect a Function's ability to send ERR_COR Messages other than the ECS SIG_SFW subclass. Default value of this bit is 0b.	RW / RO

7.9.15.4 DPC Status Register (Offset 08h)

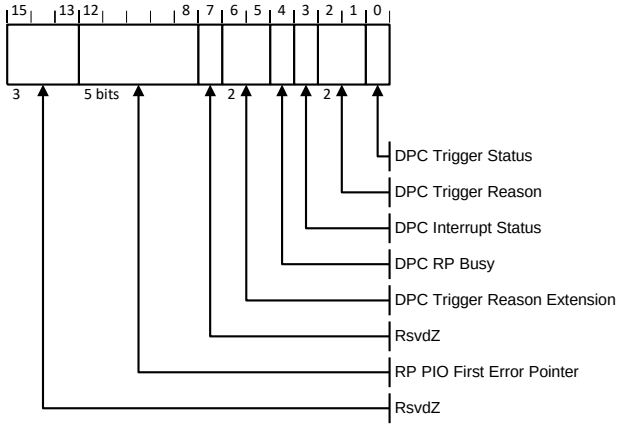

Figure 7-234: DPC Status Register

Note to Editor: Please update the preceding figure with the bit defined in the table below.

Table 7-189: DPC Status Register

Bit Location	Register Description	Attributes
…	…	…
13	**DPC SIG_SFW Status –** If the Function supports ERR_COR Subclass capability, this bit must be implemented; otherwise, it must be hardwired to 0b. If implemented, this bit is Set when a SIG_SFW ERR_COR Message is sent to signal a DPC event. See Sections 6.2.10.2 and 6.2.10.5. Default value of this bit is 0b.	RW1CS / RsvdZ

Change Section 7.9.17 Readiness Time Reporting Extended Capability as follows:

7.9.17 Readiness Time Reporting Extended Capability

The Readiness Time Reporting Extended Capability provides an optional mechanism for describing the time required for a Device or Function to become Configuration-ready. In the indicated situations, software is permitted to issue Requests to the Device or Function after waiting for the time advertised in this capability and need not wait for the (longer) times required elsewhere.

…

Software is permitted to cache values from this capability and to use those cached values ~~when~~ as long as the same device ~~topology~~ operating in the same manner has not changed.

Add new sub-section to Section 7.9 Additional PCI and PCIe Capabilities as follows:

7.9.21 SFI Extended Capability

The SFI (System Firmware Intermediary) Extended Capability is an optional capability that provides system firmware with enhanced control over primarily hot-plug mechanisms, and enables system firmware to operate as an intermediary between certain events and the operating system (see Section 6.7.4). This capability may be implemented by a Root Port or a Switch Downstream Port. It is not applicable to any other Device/Port type.

If a Downstream Port implements the SFI Extended Capability, that Port must support ERR_COR Subclass capability, and indicate so by Setting the ERR_COR Subclass Capable bit in the Device Capabilities register. See Section 7.5.3.3.

31	0	Byte Offset
PCI Express Extended Capability Header		000h
SFI Control Register	SFI Capability Register	004h
RsvdP	SFI Status Register	008h
SFI CAM Address		00Ch
SFI CAM Data		010h

Figure 7-f2: SFI Extended Capability

7.9.21.1 SFI Extended Capability Header (Offset 00h)

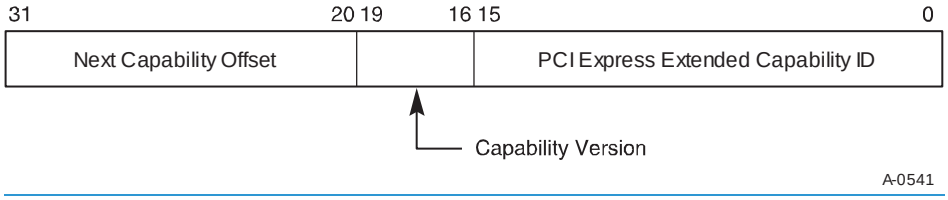

Figure 7-f3: SFI Extended Capability Header

Table 7-t2: SFI Extended Capability Header

Bit Location	Register Description	Attributes
15:0	**PCI Express Extended Capability ID** – This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. PCI Express Extended Capability ID for the SFI Extended Capability is 002Ch.	RO
19:16	**Capability Version** – This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Must be 1h for this version of the specification.	RO

Bit Location	Register Description	Attributes
31:20	**Next Capability Offset** – This field contains the offset to the next PCI Express Extended Capability structure or 000h if no other items exist in the linked list of Capabilities.	RO

7.9.21.2 SFI Capability Register (Offset 04h)

Note to Editor: Please create a figure here using standard PCIe conventions for a 16-bit RsvdP register with fields and/or bits as defined in the table below.

Figure 7-f4: SFI Capability Register

Table 7-t3: SFI Capability Register

Bit Location	Register Description	Attributes
0	**SFI OOB PD Supported** – When Set, this bit indicates that this slot supports reporting the out-of-band presence detect state. If this Downstream Port has no implemented slot (as indicated by the Slot Implemented bit in the PCI Express Capabilities register), then the value of this bit must be 0b.	HwInit

7.9.21.3 SFI Control Register (Offset 06h)

Note to Editor: Please create a figure here using standard PCIe conventions for a 16-bit RsvdP register with fields and/or bits as defined in the table below.

Figure 7-f5: SFI Control Register

Table 7-t4: SFI Control Register

Bit Location	Register Description	Attributes
0	**SFI PD State Mask –** When Set, this bit masks the Presence Detect State bit in the Slot Status register, making its value 0b, regardless of the actual presence detect state. Otherwise, its value indicates the actual state. If the value of the Presence Detect State bit changes when the SFI PD State Mask bit value changes, this must cause a Presence Detect Changed event (see Section 6.7.3). Default value of this bit is 0b.	RW

Bit Location	Register Description	Attributes
1	**SFI DLL State Mask –** When Set, this bit masks the Data Link Layer Link Active bit in the Link Status register, making its value 0b, regardless of the actual Data Link Layer state. Otherwise, its value indicates the actual state. If the value of the Data Link Layer Link Active State bit changes when the SFI DLL State Mask bit value changes, this must cause a Data Link Layer State Changed event (see Section 6.7.3). Default value of this bit is 0b.	RW
2	**SFI OOB PD Changed Enable –** When Set, this bit enables sending an ERR_COR Message for the SFI OOB PD Changed event. See Section 6.7.4.1 for other necessary conditions. This bit must be RW if the SFI OOB PD Supported bit is Set; otherwise, it is permitted to be hardwired to 0b. If the SFI OOB PD Supported bit is Clear and software Sets this bit, the behavior is undefined. Default value of this bit is 0b.	RW / RO
3	**SFI DLL State Changed Enable –** When Set, this bit enables sending an ERR_COR Message for the SFI DLL State Changed event. See Section 6.7.4.1 for other necessary conditions. Default value of this bit is 0b.	RW
5:4	**SFI DPF Control –** This field controls the level of Downstream Port Filtering (DPF) enabled on the Downstream Port, governing which Request TLPs targeting Downstream Components get filtered; that is, handled as if the Link is in DL_Down. See Section 6.7.4.2. Defined encodings are: 00b Disabled 01b Filter all Request TLPs 10b Filter only Configuration Request TLPs 11b Reserved Default value of this field is 00b.	RW
6	**SFI HPS Suppress –** When Set, this bit forces the Hot-Plug Surprise (HPS) bit in the Slot Capabilities register to be Clear and disables associated Hot-Plug Surprise functionality. See Section 6.7.4.4. Default value of this bit is 0b.	RW
7	**SFI DRS Mask –** When Set, this bit masks the DRS Message Received bit in the Link Status 2 register, making its value 0b, regardless of the actual DRS Message Received state. Otherwise, its value indicates the actual state. If the value of the DRS Message Received bit changes from Clear to Set when the SFI DRS Mask bit is Cleared, this must trigger any notification enabled by the DRS Signaling Control field in the Link Control register (see Section 7.5.3.7). Default value of this bit is 0b.	RW
8	**SFI DRS Signaling Enable –** When Set, this bit enables sending an ERR_COR Message for the SFI DRS Received event. See Section 6.7.4.1 for other necessary conditions. Default value of this bit is 0b.	RW

Bit Location	Register Description	Attributes
9	**SFI DRS Trigger –** If the SFI DRS Mask bit is Clear, when software writes a 1b to this bit, the Downstream Port must behave as if a DRS Message was received. Otherwise, software writing a 1b to this bit has no effect. It is permitted to write 1b to this bit while simultaneously writing updated values to other fields in this register, notably the SFI DRS Mask bit. For this case, the SFI DRS Trigger semantics are based on the updated value of the SFI DRS Mask bit. This bit always returns 0b when read.	RW

7.9.21.4 SFI Status Register (Offset 08h)

Note to Editor: Please create a figure here using standard PCIe conventions for a 16-bit RsvdZ register with fields and/or bits as defined in the table below.

Figure 7-f6: SFI Status Register

Table 7-t5: SFI Status Register

Bit Location	Register Description	Attributes
0	**SFI PD State –** This bit always indicates the actual presence detect state associated with the Presence Detect State bit in the Slot Status register, even when the value of that bit is being masked by the SFI PD State Mask bit.	RO
1	**SFI OOB PD State –** This bit indicates the out-of-band presence detect state, independent of the in-band presence detect state. This bit must be implemented if the SFI OOB PD Supported bit is Set; otherwise, it is permitted to be hardwired to 0b.	RO
2	**SFI OOB PD Changed –** This bit is Set when the value reported in the SFI OOB PD State bit is changed.	RW1C
3	**SFI DLL State –** This bit always indicates the actual link state associated with the Data Link Layer Link Active bit in the Link Status register, even when the value of that bit is being masked by the SFI DLL State Mask bit.	RO
4	**SFI DLL State Changed –** This bit is Set when the value reported in the SFI DLL State bit is changed.	RW1C
5	**SFI DRS Received –** This bit always indicates the actual state associated with the DRS Message Received bit in the Link Status 2 register, even when the value of that bit is being masked by the SFI PD State Mask bit. Clearing the SFI DRS Received bit (by writing a 1b to it) must also cause the actual state associated with the DRS Message Received bit to be Cleared.	RW1C

7.9.21.5 SFI CAM Address Register (Offset 0Ch)

Figure 7-f7: SFI CAM Address Register

Table 7-t6: SFI CAM Address Register

Bit Location	Register Description	Attributes
27:0	**SFI CAM Address –** This field specifies the target Bus, Device, and Function Numbers, along with the Extended Register Number and Register Number, in the format specified by Table 7-1.	RW

7.9.21.6 SFI CAM Data Register (Offset 0Ch)

Figure 7-f8: SFI CAM Data Register

Table 7-t7: SFI CAM Data Register

Bit Location	Register Description	Attributes
31:0	**SFI CAM Data –** When this field is read, the SFI CAM generates and transmits a Configuration Read Request on the Link below this Port. When this field is written, the SFI CAM generates and transmits a Configuration Write Request on the Link below this Port. In both cases, the target of the Configuration Request is determined by the value of the SFI CAM Address register. See Section 6.7.4.3.	RW

Add new Appendix x Async Hot-Plug Reference Model as follows:

x. Async Hot-Plug Reference Model

This appendix presents a recommended reference model for async hot-plug. The reference model covers three areas:

- Async hot-plug initial configuration
- Async removal configuration and interrupt handling
- Async hot-add configuration and interrupt handling

There are no normative requirements in this section. The entire reference model is contained in a series of implementation notes. The reference model documents how the various hot-plug mechanisms are envisioned be used to implement a robust asynchronous hot-plug model, but does not mandate that they be used that way.

For brevity and readability, this section uses the following acronyms:

- DLL – Data Link Layer
- DSP – Downstream Port
- FWF – firmware first
- HPS – Hot-Plug Surprise
- OOB – out-of-band
- OS – operating system
- PD – presence detect
- SFW – system firmware

The reference model covers both the HPS and DPC mechanisms. DPC is the recommended mechanism. The reference model covers FWF for DPC but not for HPS.

The reference model provides a recommended framework for DPC software to support Containment Error Recovery (CER) along with async hot-plug. The reference model does not explicitly cover CER outside of async hot-plug, but certain aspects can be leveraged for DPC support of CER when async hot-plug is not being used.

SFW may use the System Firmware Intermediary (SFI) Capability for async hot-plug, orderly removal hot-plug, or other operations. This reference model does not rely on its functionality nor cover its use.

The reference model refers to various bits or fields outlined in Section 6.7.2. For brevity, pointers to the associated registers are not replicated in this section.

IMPLEMENTATION NOTE

In-band Presence Detect Mechanism Deprecated for Async Hot-Plug

Due to architectural issues, the in-band (Physical-Layer-based) portion of the PD mechanism is deprecated for use with async hot-plug. One issue is that in-band PD as architected does not detect adapter removal during certain LTSSM states, notably the L1 and Disabled States. Another issue is that when both in-band and OOB PD are being used together, the Presence Detect State bit and its associated interrupt mechanism always reflect the logical OR of the in-band and OOB PD states, and with some hot-plug hardware configurations, it is important for software to detect and respond to in-band and OOB PD events independently.

If OOB PD is being used and the associated DSP supports In-Band PD Disable, it is recommended that the In-Band PD Disable bit be Set, and the Presence Detect State bit and its associated interrupt mechanism be used exclusively for OOB PD.

As a substitute for in-band PD with async hot-plug, the reference model uses either the DPC or the DLL Link Active mechanism.

The reference model assumes and covers the configurations listed below. While these cover the bulk of the envisioned use cases, many minor variations are not explicitly covered. For example, there are multiple ways to determine if a slot supports OOB PD, and the reference model does not cover them. As another example, the reference model refers to adding or removing an adapter, but some hot-pluggable modules may include a Switch and multiple Endpoint components.

- Reference model assumptions:
  - DSPs support DLL Link Active Reporting
  - DSPs support In-Band PD Disable
  - Operating systems support both HPS and DPC, using DPC if available
- Reference model covers:
  - Slots that support HPS and/or DPC (SW never enables both at same time)
  - Slots with or without OOB PD
  - RPs with or without RP Extensions for DPC

x.1 Async Hot-Plug Initial Configuration

IMPLEMENTATION NOTE

Async Hot-Plug Initial Configuration

Basic steps	HPS	DPC
Determine capability control entity	OS requests, and is granted control of PCIe Native Hot-Plug If FWF, SFW retains control of AER and DPC Else OS requests and is granted control of AER and DPC	
OS and SFW determine which async hot-plug mechanism to use; OS/SFW interactions here are outside scope of this specification	❑ If DPC capability then ❑ If HPS bit not Set, use DPC ❑ Else attempt to Clear HPS bit (Section 6.7.4.4) ❑ If successful, use DPC ❑ Else use HPS ❑ Else if HPS bit Set, use HPS ❑ Else async hot-plug cannot be supported by this slot ❑ Configure DSP for selected mechanism	
OS determines if adapter present	❑ If OOB PD supported, use it to determine if adapter is present ❑ Set In-band PD Disable bit (SFW may have it Set by default) ❑ Read PD State bit ❑ Else allow Link to attempt to train; use DLL Link Active to determine if adapter is present (and at least minimally functional)	
If adapter is present, OS waits for adapter to become ready for configuration	❑ If using Device Readiness Status (DRS), begin configuration if/after DSP receives a DRS Message ❑ If using CRS Software Visibility, can attempt first Configuration Read (of Vendor ID field) after 100 ms ❑ Otherwise, must wait at least 1000 ms before attempting first Configuration Read	
OS configures for appropriate case	❑ If adapter is present, configure system for handling async removal ❑ Else configure system for handling async hot-add	

x.2 Async Removal Configuration and Interrupt Handling

IMPLEMENTATION NOTE

Async Removal Configuration

Basic steps	HPS	DPC
OS/SFW configure appropriate error handling	❑ If OS and driver support a non-CER error recovery approach, its policies may influence some of these error settings ❑ Configure the error handling supported by HPS	❑ If OS and driver support CER, its policies may influence some of these error settings ❑ Enable uncorrectable AER errors to trigger DPC, including Surprise Down ❑ If using RP Extensions for DPC, configure RP PIO error handling ❑ Configure RP Completion Timeout handling per platform and OS policies
OS/SFW configure async removal interrupts	❑ OS enables DLL State Changed interrupt	❑ If FWF, configure DPC for ERR_COR signaling to enable SFW handling ❑ Else configure DPC for interrupt to enable OS handling
	❑ If OOB PD supported, OS enables PD Changed interrupt	

IMPLEMENTATION NOTE

Async Removal Interrupt Handling

Basic steps	HPS	DPC
Service routine entry	❑ If PD Changed or DLL State Change, OS is interrupted	❑ If PD Changed, OS is interrupted ❑ If FWF, OS invokes SFW; preferably via OS Setting DPC Software Trigger bit (if implemented)
		❑ If DPC triggered ❑ If FWF, DPC sends ERR_COR to signal SFW ❑ Else DPC interrupts OS
Prevent further Link activity	❑ OS Sets Disable Link; Link goes down and stays down ❑ OS polls DLL Link Active until Clear	❑ DPC automatically keeps Link down
Log errors from DSP's AER/DPC		❑ If FWF, SFW logs DSP errors ❑ SFW invokes OS and exits
	❑ OS logs and then clears DSP errors	
OS notifies impacted software/driver, which cease accessing the adapter		
OS determines if adapter is still present	❑ Some FFs with OOB PD automatically power off slot and/or disable switched signals ❑ If OOB PD supported, use it to determine if adapter is physically present ❑ If adapter not determined to be absent, re-enable the Link and slot ❑ As applicable, Clear Disable Link or release DPC ❑ Wait until Link trains or adapter is deemed absent or non-functional ❑ If non-functional, optionally perform a slot reset (if supported) ❑ If still non-functional, optionally power-cycle the slot (if supported)	
If OS determined adapter to be present	❑ OS waits for adapter to become ready for configuration ❑ OS enumerates and configures the adapter ❑ If DPC and FWF, OS invokes SFW to log adapter AER errors ❑ OS logs adapter AER errors and then clears them ❑ If OS determines the adapter is suitable for continued operation ❑ OS configures for async removal handling ❑ OS resumes driver ❑ Else OS takes OS-specific action	
Else (adapter not present)	❑ OS ensures DSP is in a clean state ready for a new/different adapter ❑ OS configures for async hot-add handling	

x.3 Async Hot-Add Configuration and Interrupt Handling

IMPLEMENTATION NOTE

Async Hot-Add Configuration

Basic steps	HPS	DPC
OS configures for async hot-add handling ❑ Enable Link to train if/when an adapter is inserted ❑ E.g., Clear Disable Link or release DPC if needed ❑ If appropriate for form factor, enable power controller prior to adapter insertion ❑ If OOB PD supported, enable OS interrupt on PD Changed ❑ Else enable OS interrupt on DLL State Changed		

IMPLEMENTATION NOTE

Async Hot-Add Interrupt Handling

Basic steps	HPS	DPC
OS is interrupted due to PD Changed or DLL State Changed		
OS waits for the adapter to become ready for configuration ❑ If appropriate for form factor, enable power controller now (following adapter insertion) ❑ Wait for DLL Link Active ❑ Wait for adapter to become ready for configuration		
OS configures adapter per standard OS conventions		
OS configures for async removal handling		
OS calls driver to complete adapter configuration and begin normal operation		